Automated Door Attendant : Speech Integration

Ali-Akber Saifee
Department of Electrical and Computer Engineering,
McGill University
3630 University Street
Montreal, Quebec H3A 2B2
ali.saifee@mail.mcgill.ca

Abstract – The Automated Door Attendant was designed to function as a multi-modal virtual secretary system for use by professors. This paper will attempt at outlining the process of developing and integrating the speech interaction module for this system.

I. Introduction

The current development of the Automated Door Attendant has been underway since 2002 at the Centre for Intelligent Machines at McGill University. Areas of development that have been concentrated on, include usability improvement, broadening the functionality of the system and improving it's stability [1].

There are two different interfaces to ADA; the professor side interface and the user side interface. Speech interaction is essentially more useful towards the user side interface – and is therefore being developed solely for this purpose [1].

The scope of speech interaction required of ADA takes two distinct forms. The first, requires speech recognition, in order to allow the user to communicate their desired actions in the system. The second, simply requires speech capture, so that users may leave messages for the professor. The latter requires no recognition - as the speech is simply relayed in it's original form to the professor – and is already a functional part of ADA. The former, however, is the proponent of ADA that will be the topic of this paper.

II. Design Goals

Before any concrete steps towards the design and implementation of the speech interaction for ADA could be made it was important to place the goals of this project into context.

The first factor of importance was the fact that this development was to be the basis of experimentation and testing of human user interaction with computer augmented systems – through speech. Therefore, the resultant speech interaction module needed to be aimed towards a testable; if not completely stable system.

Because this system would inherently be involved in various tests to further expand the knowledge base regarding human-computer speech interaction trends, it was necessary to ensure that the design could be expanded upon and easily altered in the future. This goal was further strengthened by the fact that user testing would undoubtedly uncover many aspects of interaction which had not been considered during the initial design stage.

In order to accomplish the above mentioned, it was decided to enforce two, more specific design goals, i.e., the need for modularity of design and unit testing. Both these goals would ensure that a certain level of dependability could be placed upon the speech interaction module – and this dependability could be modified and improved upon without affecting the parallel development and testing of the graphical user interface.

III. Evolution Of Design

A. Speech Recognition Engine

The first step towards developing the speech interaction module for ADA was to select a suitable software speech recognition engine. The factor that was most important towards the choice of speech recognition engines, was ensuring a high degree of accuracy. Depth in terms of vocabulary was not very important as the scope of conversation with ADA is quite limited and does not require a very extensive vocabulary.

After an initial evaluation of the speech recognition engines that were available, the most viable option was deemed to be the CMU Sphinx-4 Speech Recognition System, that is being developed at the Carnegie Mellon University in conjunction with Sun Microsystems Laboratories, Mitsubishi Electronics

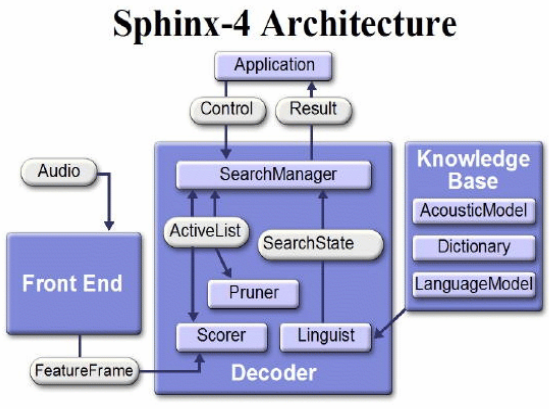

Fig.1 Architecture of the Sphinx-4 Speech Recognizer

Research Lab, and Hewlett-Packard's Cambridge Research Labs.

The most attractive aspect of the Sphinx Speech Recognition System was it's high flexibility and modifiability in terms of it's front-end. Furthermore, in order to accomplish the tasks required of the speech side of ADA, modifications and customizations could be made without affecting the underlying architecture of decoder in Fig.1.

The only modifications that had to be made to the Sphinx4 System were made to the Front End and the Knowledge Base in Fig.1. The Front End had to be modified in order for it to be correctly integrated with the rest of ADA, and so that the customizations relevant to the speech recognition required by ADA could be made. The modifications relevant to the integration are further explored in the next section.

The Knowledge Base also required minimal modifications. The Dictionary and Acoustic Model that were part of the Sphinx4 system were ample for capturing speaker independent speech recognition. The Language Model, however, had to be designed so as to constrain the speech recognition engine's "search space" to interaction that was only relevant to ADA. A JSGF (Java Speech Grammar Format) representation of this language model is presented in Fig.2.

The choice to represent the vocabulary in JSGF was made due to the almost instantaneous modifiability of this format, given the developmental and experimental status of the speech recognition side of ADA. Therefore, any future modifications that need to be made to this grammar will require little effort. Another useful aspect of utilizing the JSGF format – which resembles the Backus Naur Form, often used to define linguistic syntax – is the ease with which a large range of utterances can be defined in a logical manner. Empirical calculations have shown the depth of the word-list to be approximately 90 distinct words, and the depth of the possible utterances to be a factor of 5K.

```
grammar ada;
public <command> = ([<start>]<verbNoun> | [[<start>]<makeVerb> ([an]
appointment | [a] meeting) [with Jeremy](for | on | at)]<dayNtime>);
<start> = (I would like to | I'd like to | I want to | may i | can i );
<seeVerb> = ( see | view | look at );
<makeVerb> = ( make | schedule | set | choose | confirm);
<messageVerb> = (start | stop | replay | erase | re record | record | write | leave
| confirm );
<otherVerb> = (do | have );

<verbNoun> = ([<seeVerb>][a | the | this | Jeremy [is] ] schedule |
[<makeVerb>][an | a | this | the ] (appointment | meeting ) [with Jeremy] |
[<otherVerb>] something else |[<seeVerb>] [ the ] (following | next ) (week |
week's) [schedule] |[<seeVerb>][ the ] (previous | last | this ) (week | week's)
[schedule]|[<messageVerb>] [ a | the | this | my ] message [for Jeremy] |
[<messageVerb>][ a ] written message [(to | for ) Jeremy ] |[<messageVerb>]
[ a | the | this | my ] video message [(to | for ) Jeremy] | [<makeVerb>](another
| a different | some other ) time |[<messageVerb>] [ a | the | this ] recording |
[<seeVerb>] [the available | the | a ] documents);

<dayNtime>=<day> <time> | <time> [for | on ] <day>;
<day>=(monday | tuesday | wednesday | thursday | friday);
<time> = [half past ] (eight | nine | ten | eleven | (twelve | noon) | one | two |
three | four | five ) [ oh clock | thirty ];
```

Fig.2 JSGF Grammar used for Speech Recognition in ADA

As denoted in Fig. 2, the user is allowed to input an utterance that can range between a single-word command, and a completely structured sentence. Due to the 'and/or/maybe' structure of the grammar, partial sentences are also accepted by the system.

This ensures that the system does not provide constraints to the user in terms of the structure of the sentence – but only by the requirement of the operative instruction. Arguably, users can communicate via unpredictable variances from the knowledge base of the Speech Recognition System – however, the structure of the JSGF grammar accommodates these occurrences by looking for the operative instruction (e.g. 'message'). As will be explained further in the next section, the Speech Recognition Engine is not expected to ensure intelligent interpretation of the user's input, but only to ensure correct recognition. The necessity of this requirement can be exemplified by a user saying, "I really need to make an appointment with the Professor," and the system only recognizing "make an appointment with the Professor". The recognized sentence is still correct as all important and useful information has been extracted from the user's input. It is indeed almost impossible to construct a 'complete' language model, even within a certain context, however, a sufficient model can be constructed by isolating the necessary operative syntax from the language [5].

B. Integration of ADA with Speech Recognition Engine

The Speech Recognition Engine was not given the task of interpreting the 'meaning' of the user's speech input – but only to recognize this input. The preset requirement of modularity enforced the isolation of this task into a separate module. This resulted in the evolution of the design now implemented in the form of Fig. 3. This design ensures that minimal alterations have to be made to the individual Sphinx-4 and ADA modules, and the interaction between the two is largely handled by the intermediary module Via. The only modifications that were made to the Sphinx4 were additions

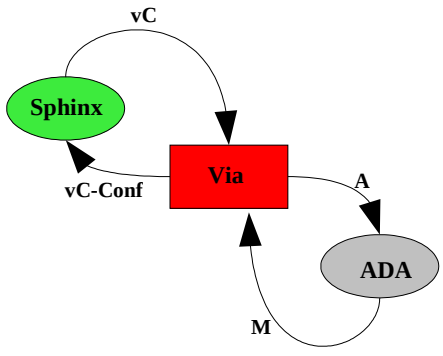

Fig. 3 Design used to integrate ADA with the Speech Recognition Engine

TABLE I
MESSAGES USED FOR SPHINX-VIA-ADA INTERACTION

Message	Purpose
vC	Conveys the recognized speech segment to VIA.
vC-Conf	Confirmation for reception of vC. Also serves as a miscellaneous communicator between Via-Sphinx4 in case Via needs Sphinx4 to stop 'listening.' etc...
M	Conveys the current mode/state of ADA.
A	Conveys the action that corresponds to the given *M* and *vC* at any given time.

of function calls to send the message vC and to receive the message vC-Conf. This cyclical message sequence was implemented using a Server/Client TCP architecture. The purpose of each message is explained in more detail in Table I.

Having understood the functions of these four messages, the underlying function of Via must become clearer to the reader. The core features of Via include message communication, synchronization and ensuring intelligent speech interpretation. Via is therefore the portion of the Speech module that makes sense of what the user has said by contextually analyzing the input.

The modifications made to ADA (to facilitate the interaction) were slightly higher in complexity than the ones made to Sphinx4. The addition of function calls to send the message M and receive the message A were the first modifications to the functionality of ADA.

TABLE II
LISTING OF THE POSSIBLE STATES OF ADA

M:=
0 -*STATE_MAINMENU*
1 -*STATE_SCREENSAVER*
2 -*STATE_SCHEDULE_THISWEEK*
3 -*STATE_SCHEDULE_NEXTWEEK*
4 -*STATE_SCHEDULE_SUGGEST*
5 -*STATE_SCHEDULE_MANUALSELECT*
6 -*STATE_SCHEDULE_APPOINTMENT_NOTEPAD*
7 -*STATE_SCHEDULE_APPOINTMENT_RECORD*
8 -*STATE_SCHEDULE_APPOINTMENT_RECORDING*
9 -*STATE_SCHEDULE_APPOINTMENT_STOPPED*
10-*STATE_SCHEDULE_APPOINTMENT_REPLAYING*
11-*STATE_SCHEDULE_APPOINTMENT_CONFIRM*
12-*STATE_MESSAGE_SELECT*
13-*STATE_MESSAGE_NOTEPAD*
14-*STATE_MESSAGE_RECORD*
15-*STATE_MESSAGE_RECORDING*
16-*STATE_MESSAGE_STOPPED*
17-*STATE_MESSAGE_REPLAYING*
18-*STATE_MESSAGE_CONFIRM*
19-*STATE_DOCUMENTS_VIEW*
20-*STATE_VIDEOCONF_START*

In order to use these messages intelligently two more additions had to be made to ADA. The first was an implementation of a 'knowledge of state' in ADA which corresponded with the message 'M' that is sent to Via (Table II). The second addition was to allow the performance of a 'voice command' when issued by the user. This was essentially a simple function which would correlate the message 'A' received by ADA and call the corresponding GTK callback previously implemented in ADA for touch screen interaction.

IV. FUTURE CONSIDERATIONS

Functionality and usability expansions that can be made towards the Speech side of ADA are only limited by continuous design efforts, user testing and the quality of the Speech Recognition. There are however some specific considerations that could be made towards improving the performance of ADA.

The first and foremost of these is the need for user testing in order to evaluate the effectiveness of allowing for a multi-modal interface. It has been noted in [3], that users will not always perform their required actions multi-modally, but instead prefer one mode over the other for specific tasks. User testing will certainly uncover the nature of these tasks and allow for creating rigid speech recognition for these specific commands.

Another area of consideration is to explore the possibilities of reducing redundancy of content between the Speech interaction and the Touch Screen interaction. Speech contains information about the users that cannot be conveyed through gestural input. Extracting and realizing the nature of this information will certainly allow for many enhancements in ADA. An example of this kind of possibility is to utilize larger speech utterance to extract multiple commands issued by the user. This is a more natural form of communication as far as the user is concerned and a more multi-modal system as opposed to a dual-input system.

The modular separation of tasks essentially realizes a major future possibility of isolating the Sphinx, Via and ADA modules (which are also processes) onto separate microprocessors. This would lead to the eventuality of implementing an embedded system instead of a background desktop processor fulfilling the computational requirements.

V. CONCLUSION

The first development of the Speech Recognition Side of ADA has resulted in a system that is usable in terms of further evaluating and testing user interaction with a virtual secretary through speech. Testing has shown the overall accuracy of the speech interpretation to be between 70-80%. This will inherently be improved with additional user testing – followed by modifications of the knowledge base that is used to

recognize the spoken utterances. There is however, no doubt that the range of natural resources that are intuitively called upon when interacting with a virtual secretary, includes speech [4]. The challenge now, is to ensure an optimal utilization of this resource on the user's side when interacting with ADA.

ACKNOWLEDGMENTS

The author would like to thank Michael Perez for the assistance and documentation provided concerning the current development and functionality of ADA. Furthermore, the invaluable encouragement, creative insight and general support that was extended by Frank Rudzicz must be acknowledged. Lastly, the author would like to thank Professor Jeremy Cooperstock for overlooking the project and providing this opportunity.

REFERENCES

[1] M. Perez, *Redesigning the Automated Door Attendant,* Centre for Intelligent Machines, Montreal, Canada:2004.
[2] J. L. Gauvin, J. J. Gangolf, L. Lamel, *Speech Recognition for an Information Kiosk,* Proceedings of International Conference on Spoken Language Processing '96 (Philadelphia, 1996), Spoken Languages Processing Group, Orsay, France: 1996.
[3] S. Oviatt, *Ten Myths of Multimodal Interaction*, Communications of the ACM (November 1999 / Vol.42 No.11),Oregon Graduate Institute of Science and Technology, Beaverton: 1999.
[4] W.Buxton, *Speech, Language & Audition,* Chapter 8 in R.M. Baecker, J. Grudin, W. Buxton and S. Greenberg, S. : 1995.
[5] S. Furui, *Automatic Speech Recognition and its Application to Information Extraction,*Proceedings of the 37th conference on Association for Computational Linguistics (College Park, 1999), Tokyo Institute of Technology, Tokyo, Japan : 1999.